

### **Listing of the Claims**

1. (Previously presented) A method for protecting a sequence of computer instructions comprising:

- preparing first obscuring instructions having associated identification codes;
- serializing the sequence of computer instructions;
- transforming a first set of the obscuring instruction identification codes associated with some or all of the first obscuring instructions to generate a second set of obscuring instruction identification codes;
- generating second obscuring instructions using the second set of obscuring instruction identification codes; and
- injecting the second obscuring instructions into the serialized sequence of computer instructions.

2. (Previously presented) The method of Claim 1, further comprising executing the serialized sequence of computer instructions injected with the second obscured instructions, one instruction at a time.

3. (Cancelled)

4. (Previously presented) The method of Claim 1, wherein the obscuring instruction identification codes comprise numeric values, and said generating of the second set of obscuring instruction identification codes comprises performing a mathematical transformation on the numeric values of the first set of obscuring instruction

identification codes to produce the numeric values of the second set of obscuring instruction identification codes.

5. (Previously presented) The method of Claim 4, further comprising injecting into the serialized sequence of instructions injected with the second obscuring instructions, a description of the mathematical transformation performed.

6. (Previously presented) The method of Claim 1, wherein said injecting comprises systematically injecting the second obscuring instructions with a plurality of copies of a runtime manager, forming a plurality of obscured instruction blocks, each comprising a copy of the runtime manager, one or more of the serialized sequence of instructions, and one or more of the second obscuring instructions.

7. (Previously presented) The method of Claim 6, wherein said serialized sequence of instructions, said second obscuring instructions, copies of said runtime manager, and the resulting obscured instruction blocks are in source form, and the method further comprises obscuringly compiling the obscured instruction blocks into object form, preserving the obscuration.

8. (Previously presented) The method of Claim 7, further comprising successively and recursively encrypting up to all, except a root one, of the obscured instruction blocks in object form, to form an obscured executable image having the encrypted ones of the obscured instruction blocks in object form successively nested.

9. (Previously presented) The method of Claim 8, wherein the successive and recursive encryption further comprises compressing the instructions being encrypted.

10. (Previously presented) A method for protecting data of a data file comprising:

preparing first obscuring data;

injecting second obscuring data into a plurality of locations in the data file using an automated process and the first obscuring data, to organize the data of the data file into a plurality of obscured data blocks, with each of the obscured data blocks having a portion of the data and one or more obscuring data; and

successively and recursively encrypting the obscured data blocks into a plurality of encrypted obscured data blocks that are successively nested, using a plurality of encryption keys, with at least the second inner most nested encrypted obscured data block including the inner most nested encrypted obscured data block, an encryption key used in the generation of the inner most nested encrypted obscured data block, and a portion of the data.

11. (Previously presented) The method of Claim 10, wherein the method further comprises

transforming a first set of obscuring data identification codes associated with some or all of the first set of obscuring data to generate a second set of obscuring data identification codes; and

generating the second set of obscuring data using the second set of obscuring data identification codes.

12. (Previously presented) The method of Claim 11, wherein the obscuring data identification codes comprise numeric values, and said generating of the second set of obscuring data identification codes comprises performing a mathematical transformation on the numeric values of the first set of obscuring data identification

codes to produce the numeric values of the second set of obscuring data identification codes.

13. (Previously presented) The method of Claim 12, further comprising injecting into the data file injected with the second obscuring data, a description of the mathematical transformation performed.

14.-16. (Cancelled)

17. (Previously presented) The method of Claim 10, wherein the successive and recursive encryption further comprises compressing the data being encrypted.

18. (Cancelled)

19. (Cancelled)

20. (Currently amended) ~~The apparatus of Claim 19,~~ Apparatus for protecting a sequence of computer instructions, comprising:

means for storing first obscuring instructions;

means for serializing the sequence of computer instructions; and

means for automatically injecting second obscuring instructions and a plurality of copies of a runtime manager into the sequence of computer instructions, using the first obscuring instructions and the runtime manager respectively;

wherein the means of injecting are adapted to systematically inject the second obscuring instructions and copies of a runtime manager into the serialized sequence of instructions to form a plurality of nested obscured instruction blocks.

21. (Previously presented) Apparatus for protecting a sequence of computer instructions comprising:

- an obscuring instruction bank to store obscuring instructions, each of which is identified by an obscuring instruction identification code;

- a transformation function bank to store transformation functions adapted to transform obscuring instruction identification codes; and

- a generator functionally coupled to the obscuring instruction bank and the transformation function bank to generate blocks of obscuring instructions by selecting identification codes of the obscuring instructions stored in obscuring instruction bank, and transformation functions from the transformation function bank, apply said selected transformation functions to transform the selected obscuring instruction identification codes, and employ the transformed obscuring instruction identification codes to generate additional obscuring instructions.

22. (Previously presented) A method for executing a plurality of critical instructions, said method comprising:

- loading a first executable instruction block of an executable module, the first executable instruction block having one or more of the critical instructions, and the executable module further having a plurality of nested encrypted executable instruction blocks having the remaining of the critical instructions that were generated through successive and recursive encryption, and

- executing the loaded first executable instruction block, including loading the plurality of nested encrypted executable instruction blocks having a first remainder of the critical instructions, retrieving a first decryption key from the loaded plurality of nested encrypted executable instruction blocks, decrypting the loaded plurality of nested encrypted executable instruction blocks once to recover a second executable

instruction block and a first remainder of the plurality of nested encrypted executable instruction blocks having a second remainder of the critical instructions.

23. (Previously presented) The method of Claim 22, further comprising executing the second executable instruction block, including retrieving a second decryption key from the first remainder of the plurality of nested encrypted executable instruction blocks, decrypting the first remainder of the plurality of nested encrypted executable instruction blocks once to recover a third executable instruction block and a second remainder of the plurality of nested encrypted executable instruction blocks having a third remainder of the critical instructions.

24. (Previously presented) The apparatus of claim 20, wherein the obscured instructions blocks are in source form, and the apparatus further comprises obscuration compiler means for compiling the obscured instruction blocks into a plurality of obscured instruction blocks in object form, preserving the obscuration.

25. (Previously presented) The apparatus of claim 24, wherein the apparatus further comprises encryption means for successively and recursively encrypting up to all, less a root one, of the obscured instruction blocks in object form, to form an obscured executable image having the encrypted ones of the obscured instruction blocks in object form successively nested.

26. (Previously presented) The apparatus of claim 25, wherein the encryption means includes compression means for compressing instructions being encrypted.

27. (Previously presented) The apparatus of claim 21, further comprising an injector to automatically inject the additional obscuring instructions and copies of a runtime manager into the sequence of computer instructions, to form a plurality of obscured instructions blocks, each comprising a copy of the runtime manager, one or more of the computer instructions, and one or more of the obscuring instructions.

28. (New) A program product having a plurality of programming instructions adapted to program an apparatus to enable the apparatus to practice the method of claim 1.

29. (New) A program product having a plurality of programming instructions adapted to program an apparatus to enable the apparatus to practice the method of claim 10.

30. (New) A program product having a plurality of programming instructions adapted to program an apparatus to enable the apparatus to practice the method of claim 22.